



Galak-Z, Forever: Building Space-Dungeons Organically

Zach Aikman

Lead Engineer // 17-BIT



INDEPENDENT GAMES
SUMMIT

GAME DEVELOPERS CONFERENCE®

MOSCONE CENTER · SAN FRANCISCO, CA

MARCH 2-6, 2015 · EXPO: MARCH 4-6, 2015

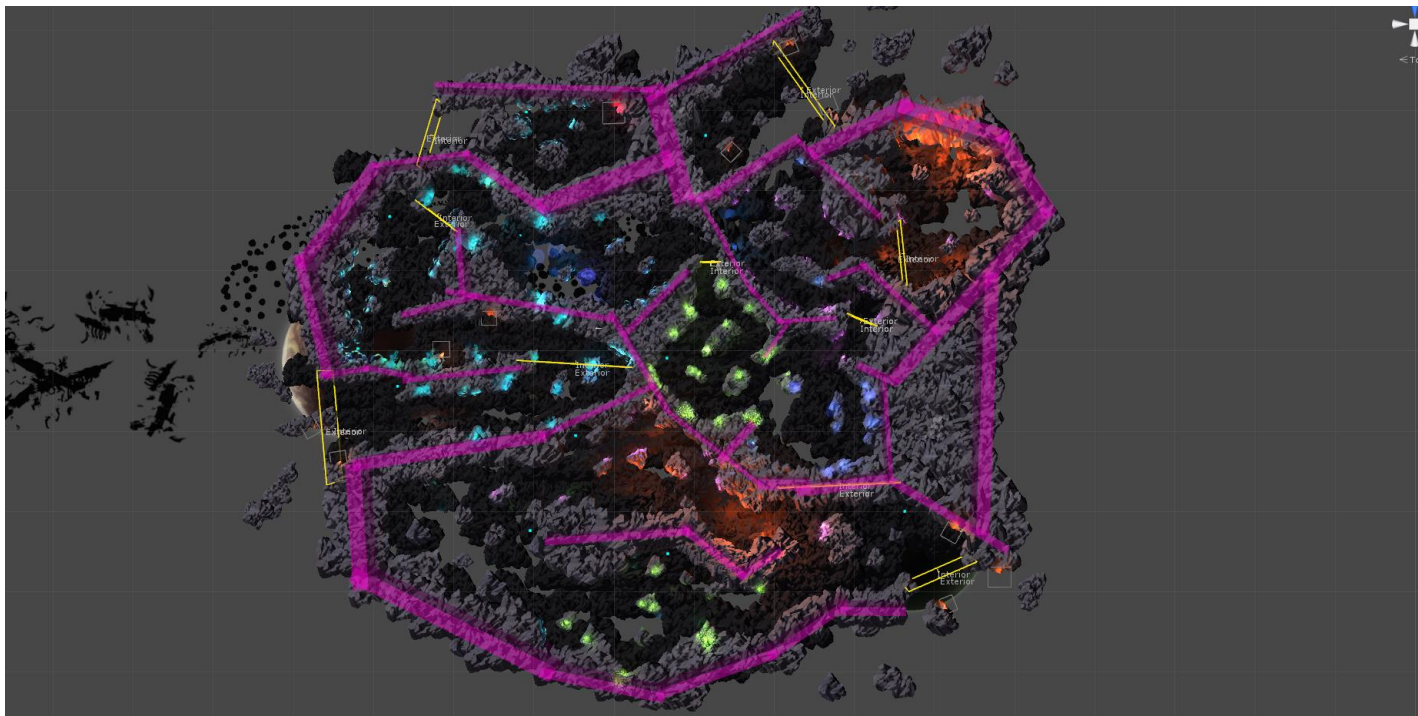


宇宙戦士 **ガラク**
GALAK-Z ★
THE DIMENSIONAL





The 4th Cave



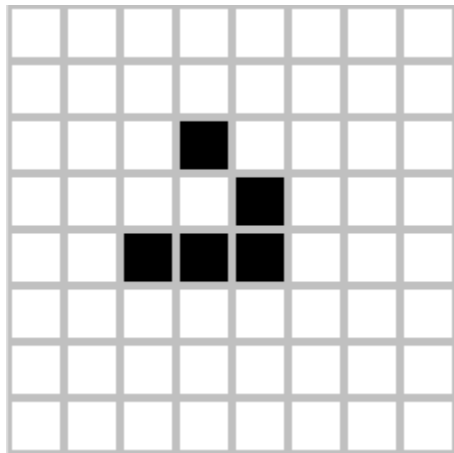


How Systems Will Save Us

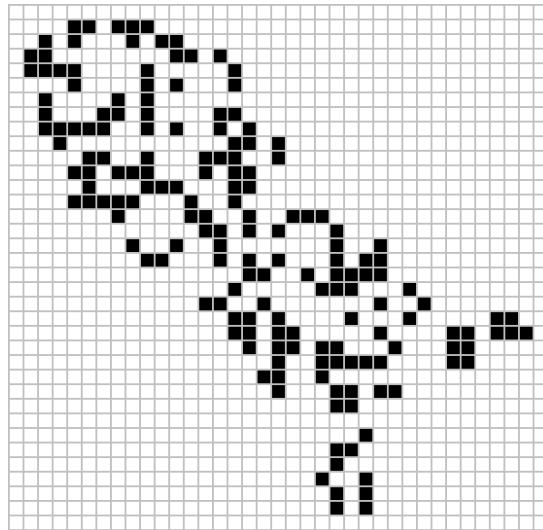
- Potential for control mastery
- Complex interaction of simple systems
- Ambience and aesthetics of dungeon regions
- Exploration of unknown spaces



Cellular Automata



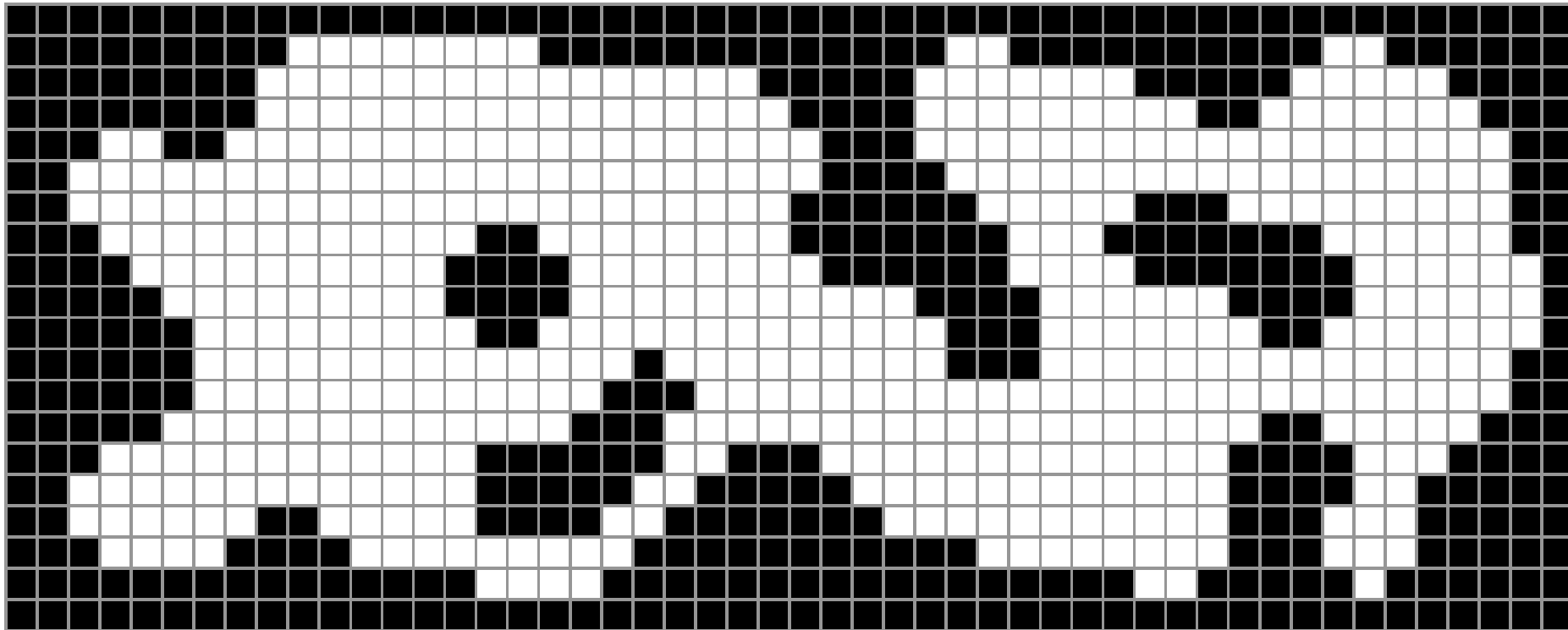
Glider



Seal

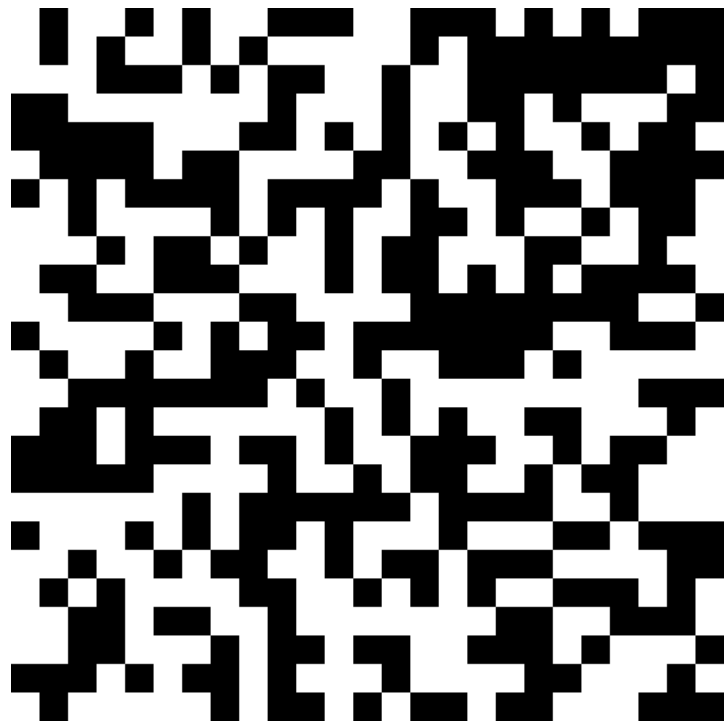


Cellular Automata



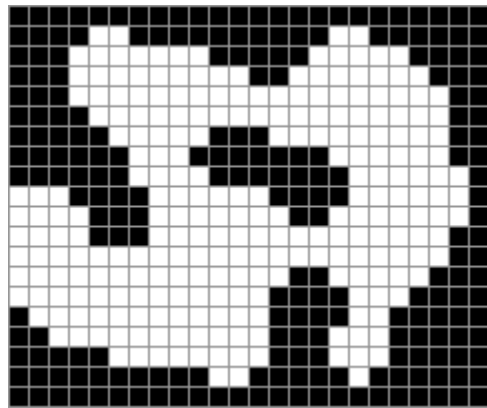
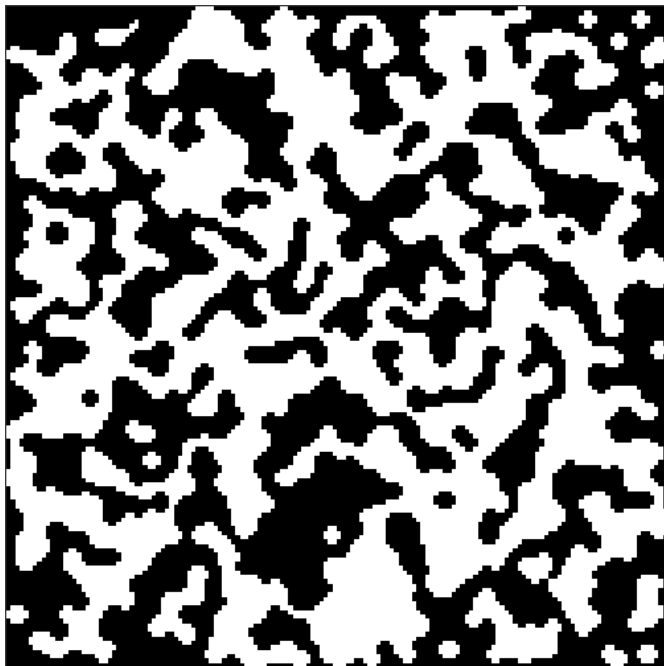


Cellular Automata



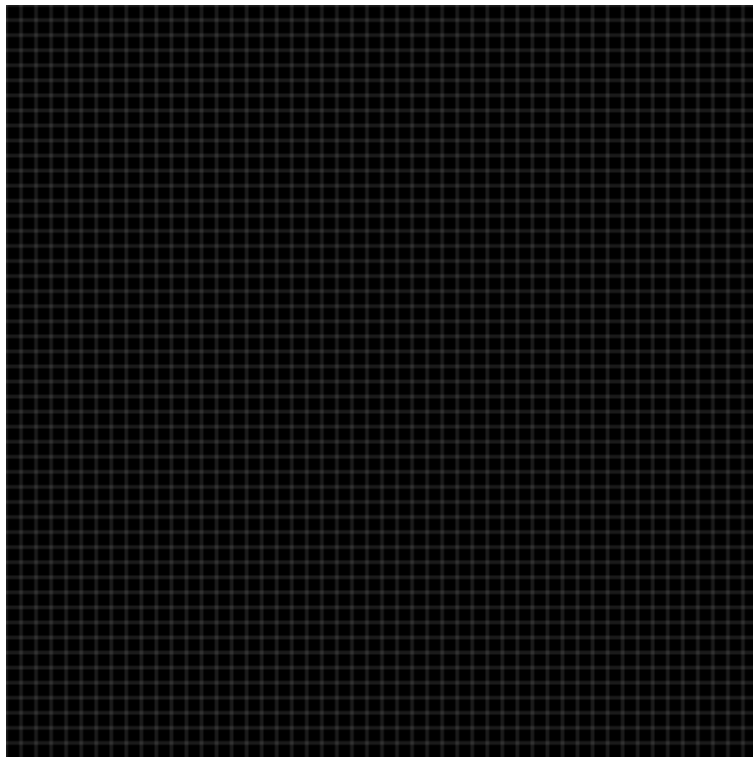


Cool! We're done, right?



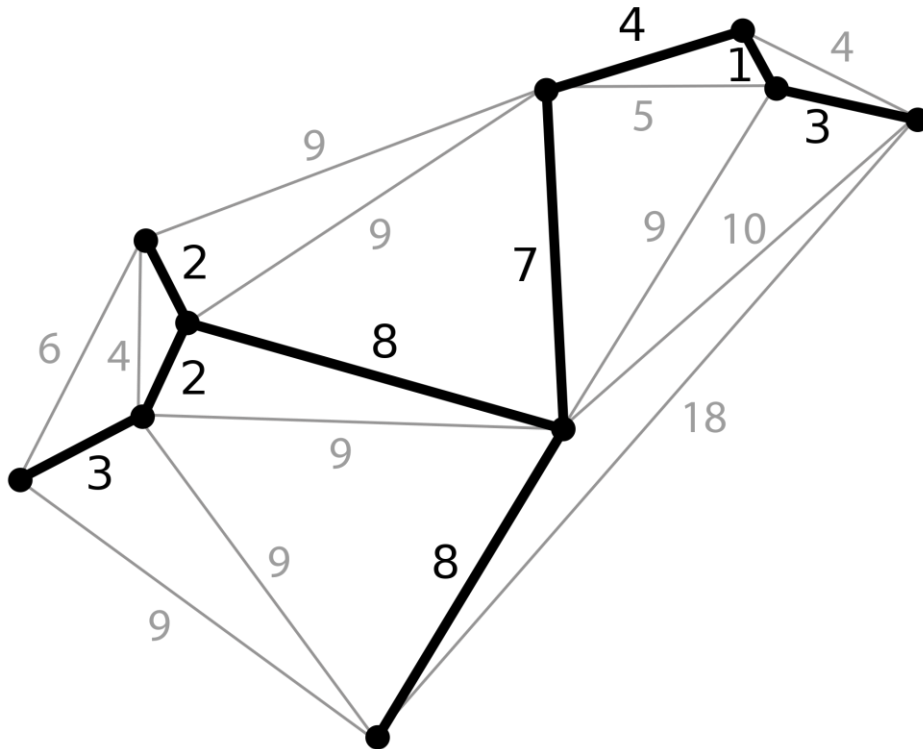


Binary Space Partitioning



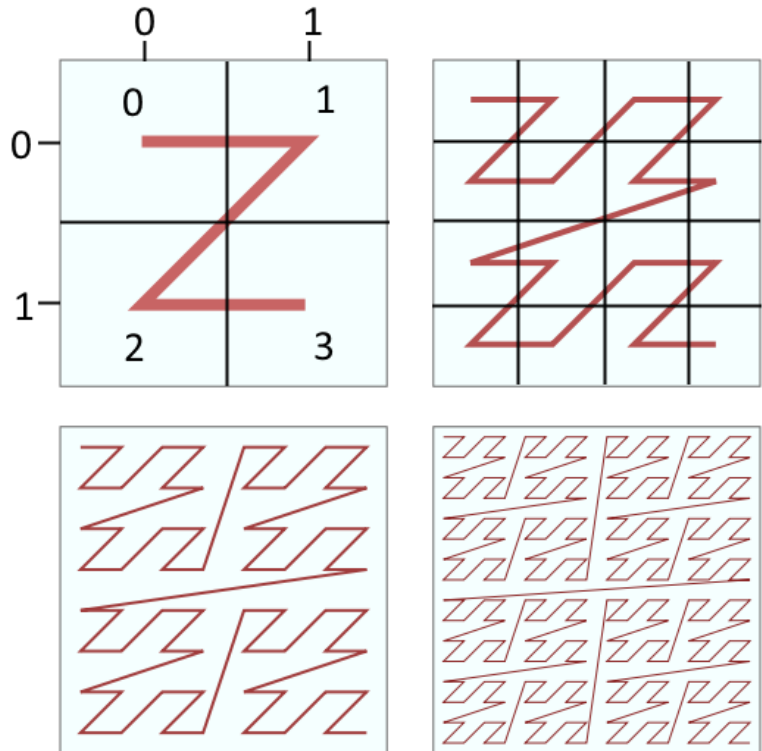


Minimum Spanning Tree





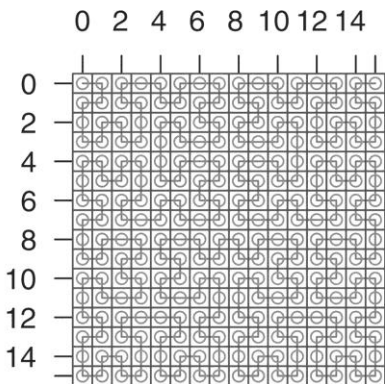
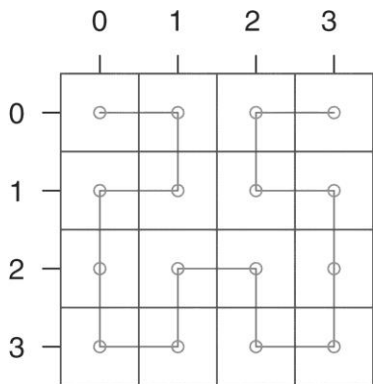
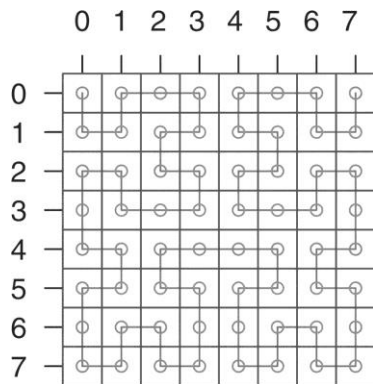
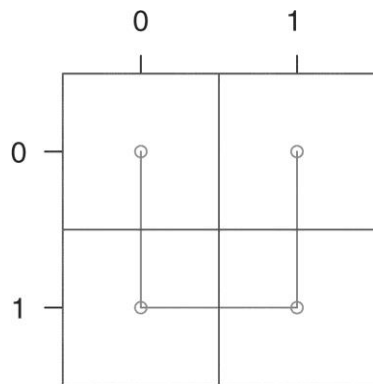
Z-Order Curves



<u>Coords</u>	<u>Z-Value</u>
$Z(0,0)$	= 0
$Z(1,0)$	= 1
$Z(0,1)$	= 2
$Z(1,1)$	= 3
⋮	
$Z(3,3)$	= 15
⋮	
$Z(7,7)$	= 63



Hilbert Curves

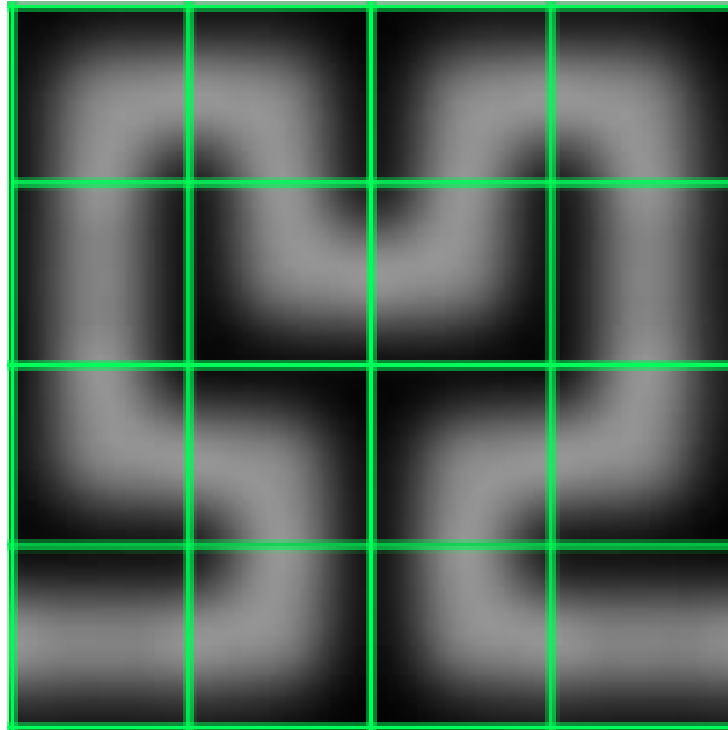


```
//convert (x,y) to d
int xy2d (int n, int x, int y) {
    int rx, ry, s, d=0;
    for (s = n/2; s > 0; s /= 2) {
        rx = (x & s) > 0;
        ry = (y & s) > 0;
        d += s * s * ((3 * rx) ^ ry);
        rot(s, &x, &y, rx, ry);
    }
    return d;
}
```

```
//convert d to (x,y)
void d2xy(int n, int d, int *x, int *y) {
    int rx, ry, s, t = d;
    *x = *y = 0;
    for (s = 1; s < n; s *= 2) {
        rx = 1 & (t / 2);
        ry = 1 & (t ^ rx);
        rot(s, x, y, rx, ry);
        *x += s * rx;
        *y += s * ry;
        t /= 4;
    }
}
```



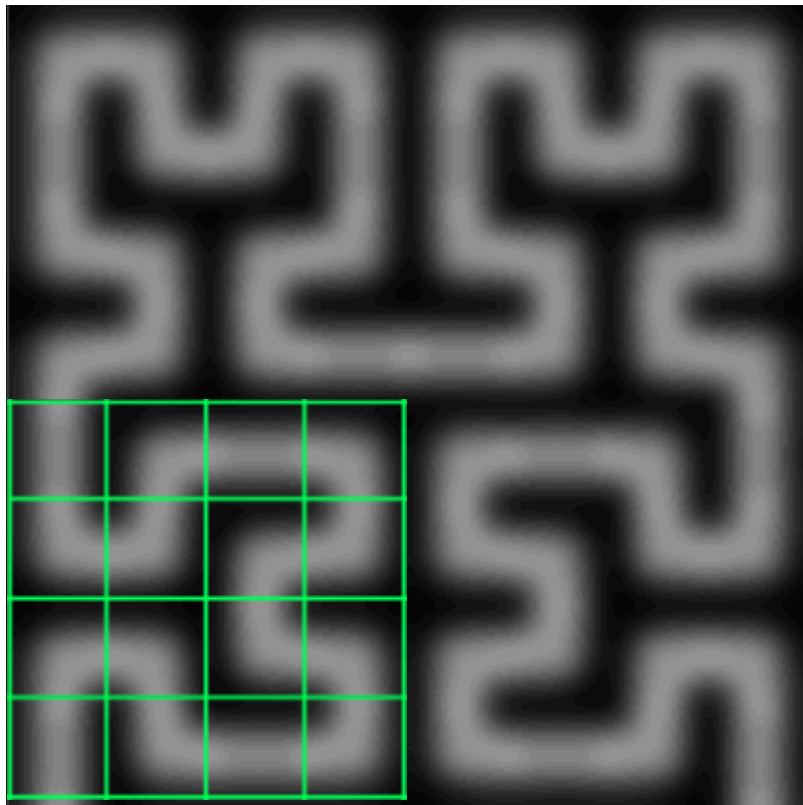
Hilbert Curves





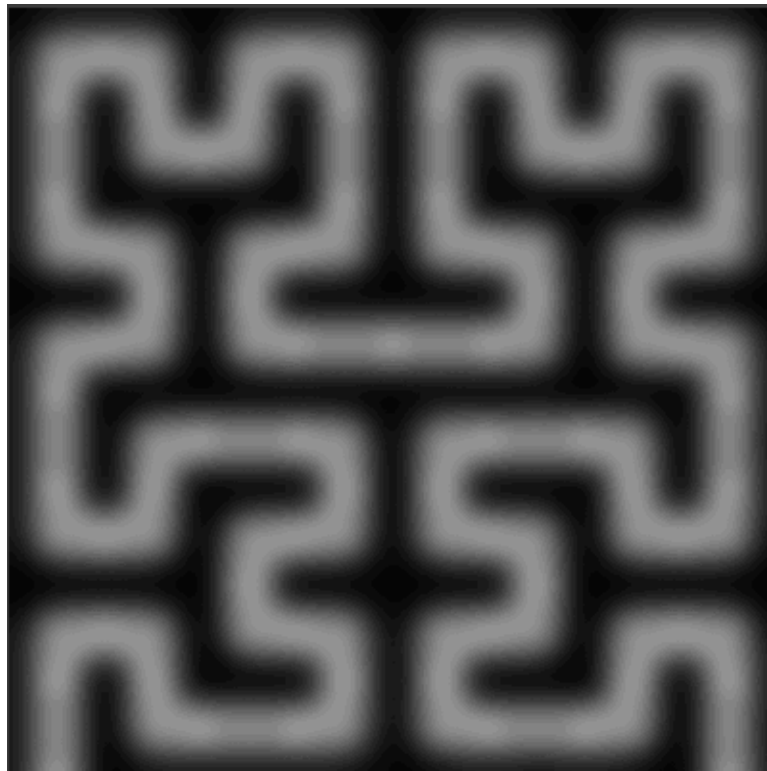
Hilbert Curves

- Increase curve dimension to next power of 2
- Randomly offset the dungeon grid
- Start at $d = 0$ and walk the curve until it lies within the grid
- If the curve goes off the grid, pick up where it comes back on
- If the curve cannot rejoin the main path when it reenters the grid, discard the lost cells





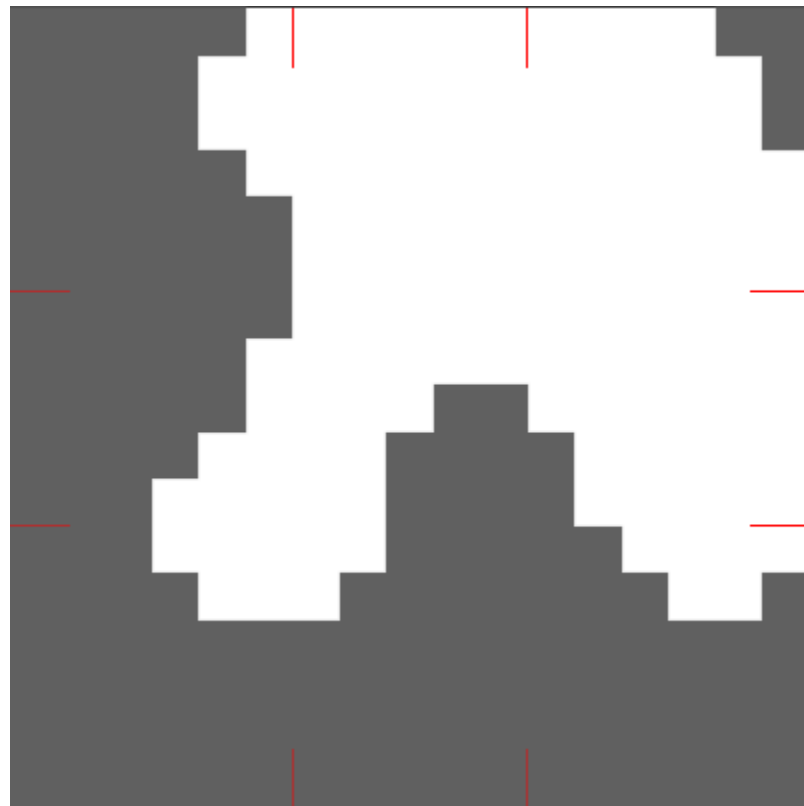
Dead Ends





Back to rooms

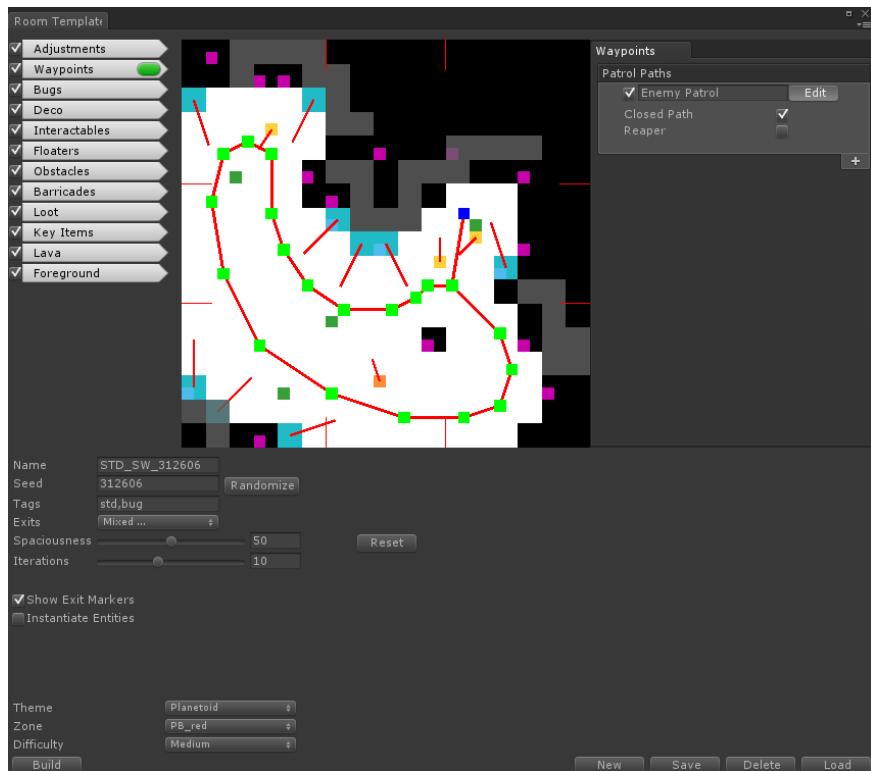
- Smaller grids (17x17)
- Fixed exits
- Terrain adjustments
- Object nodes
- Metadata





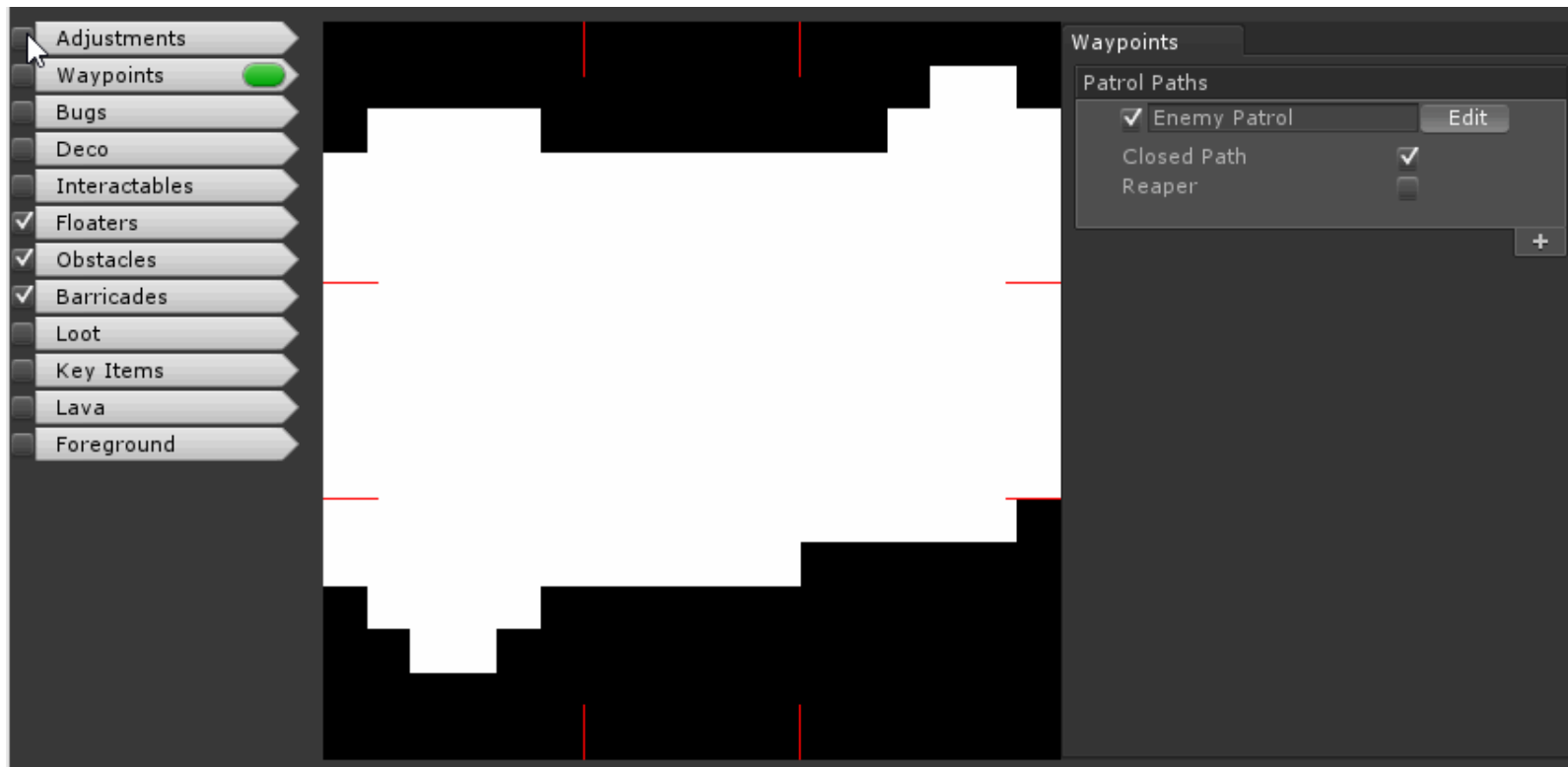
Room Template Editor

- Layer-based pixel painting
- Build button for quick previews



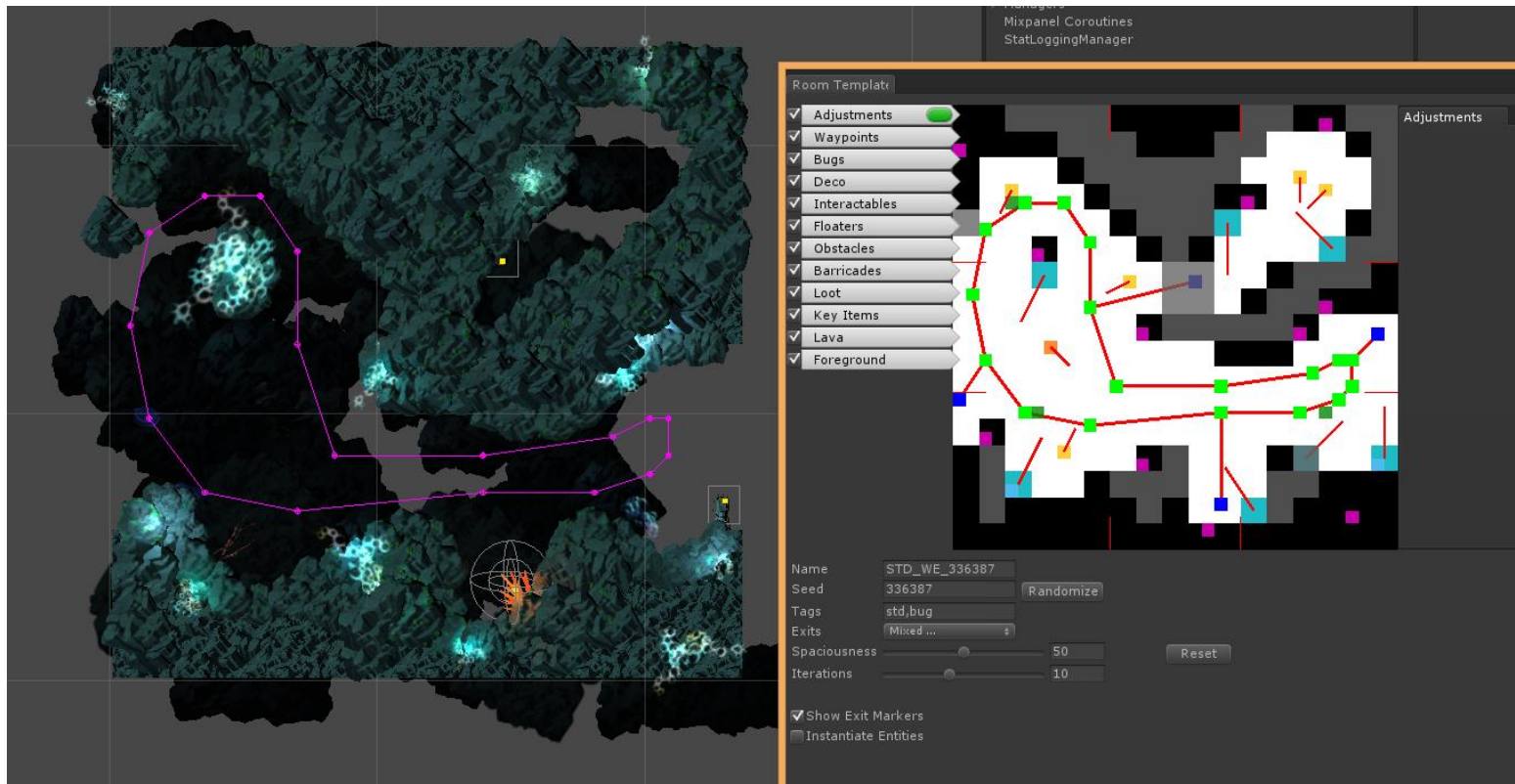


Like an onion



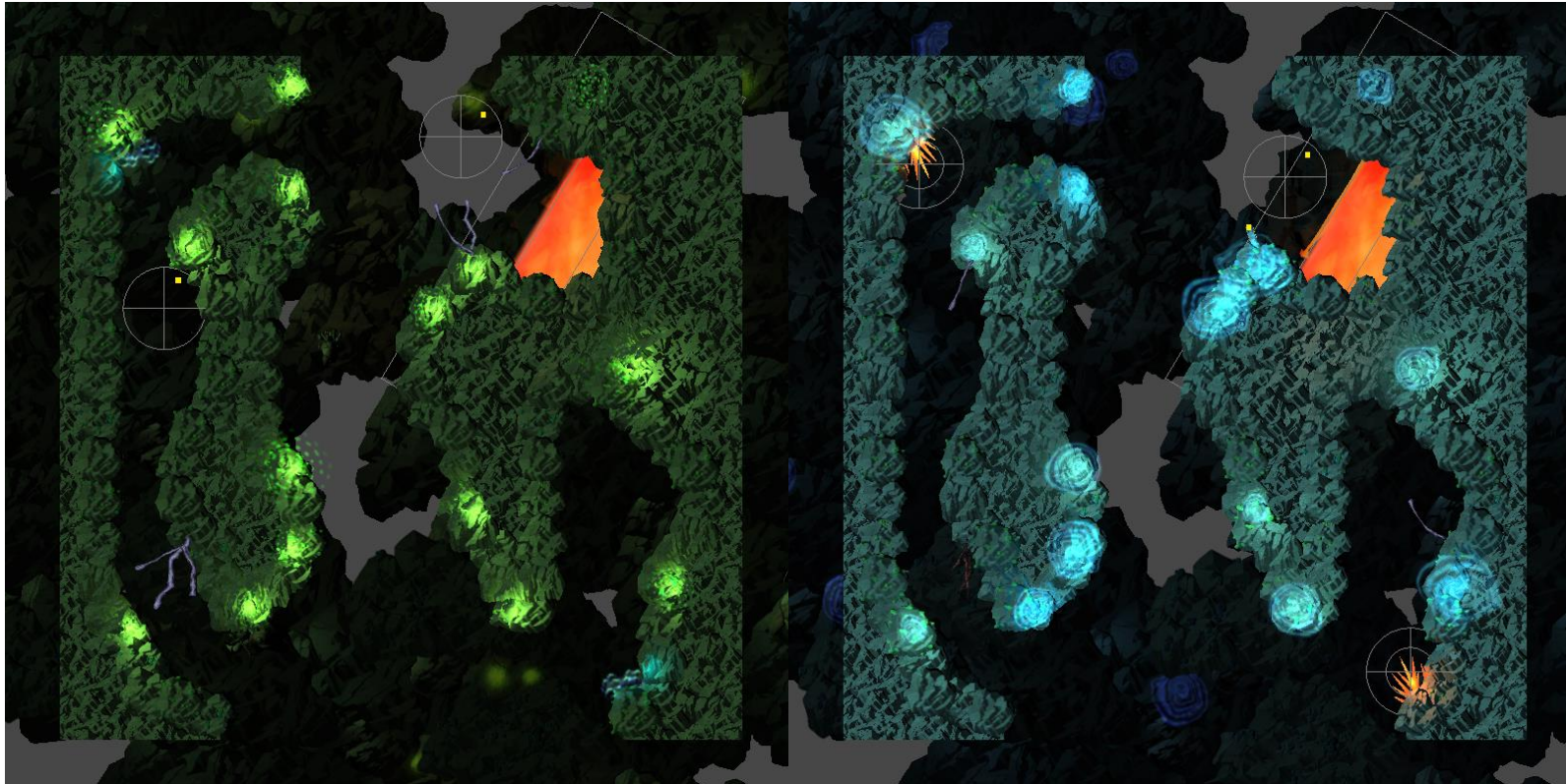


Room Previews





Dungeon Zones



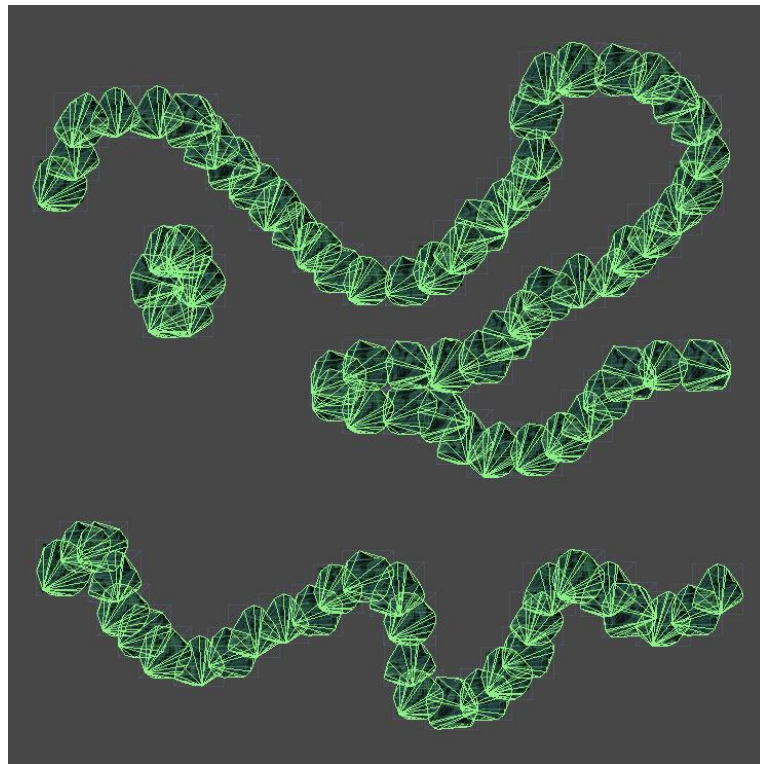
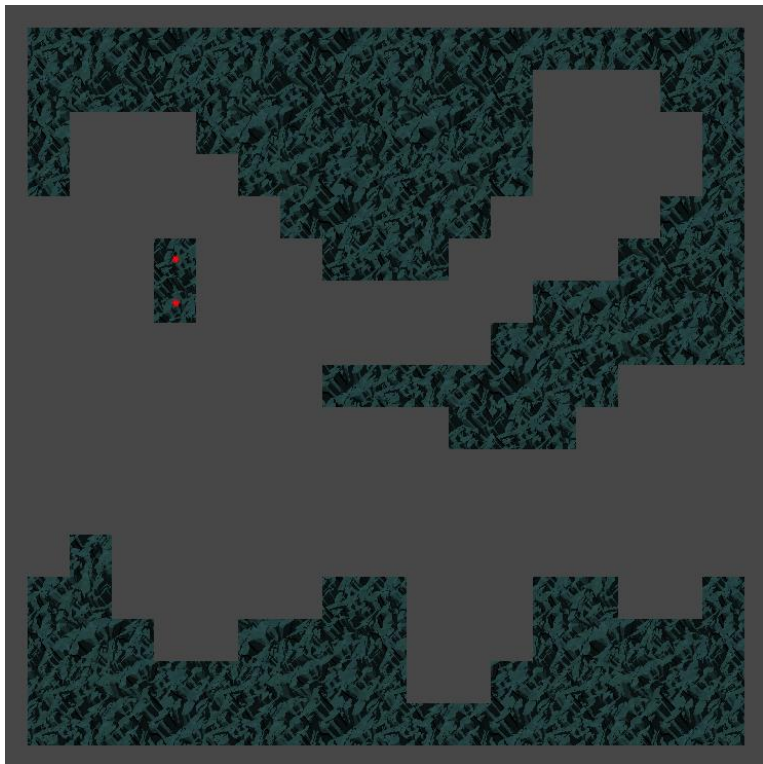


Dungeon Zones

- Directional light color / intensity
- Background deco objects
- Interactable / obstacle prefabs
- Enemy types
- Loot types
- Max room count



Blocks to rocks





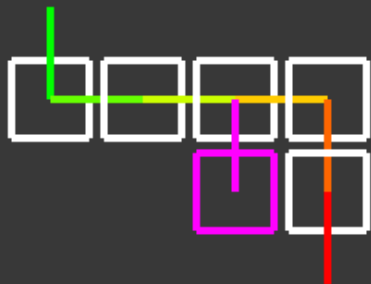
The Final Product





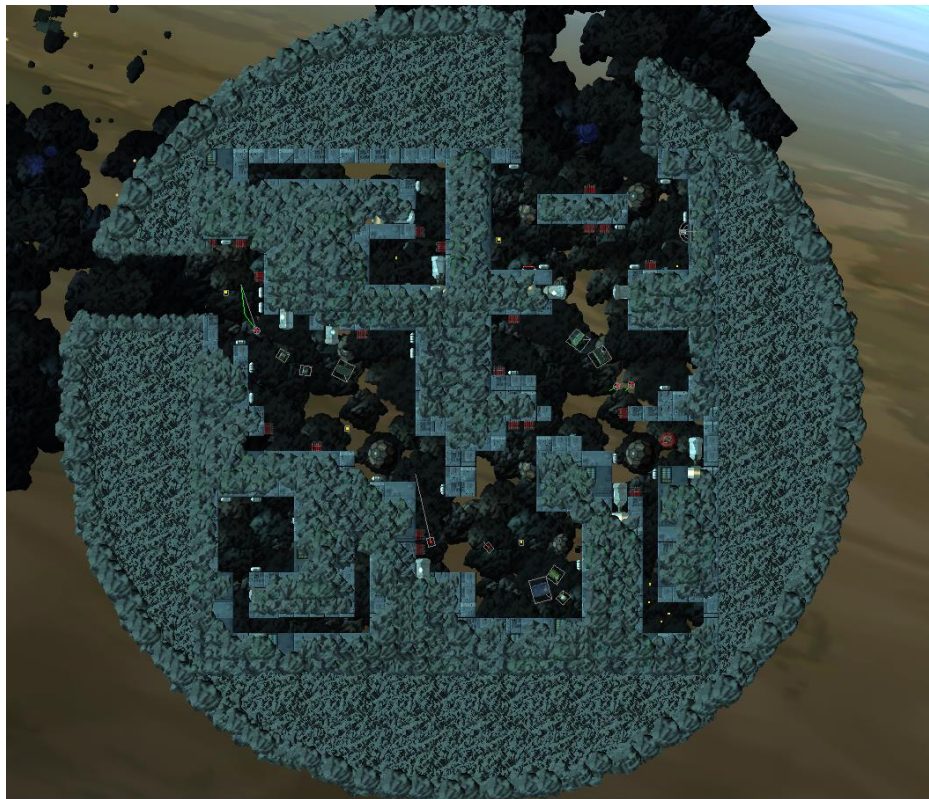
Space Hulks

0110011





Pirate Bases





Tips

- Use consistent frames of reference
- Draw pictures and keep them nearby
- Clear delegation of responsibilities



Tips

- Externalize as much data as possible
- Begin with a small, hand-crafted level
- Refactor when your working memory is exhausted
- Don't be afraid to experiment and fail spectacularly



Thank you!

Zach Aikman
@TheZaikman
zach@17-bit.com
Lead Engineer // 17-BIT